Name: John Erwin S. Ballesteros	**Date Performed:** 13/11/2024
Course/Section: CpE32s1	**Date Submitted:** 13/11/2024
Instructor: Engr. Robin Valenzuela	**Semester and SY:** 1st Sem, '24-'25

Activity 11: Containerization

1. Objectives

Create a Dockerfile and form a workflow using Ansible as Infrastructure as Code (IaC) to enable Continuous Delivery process

2. Discussion

Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications. By taking advantage of Docker's methodologies for shipping, testing, and deploying code quickly, you can significantly reduce the delay between writing code and running it in production.

Source: https://docs.docker.com/get-started/overview/

You may also check the difference between containers and virtual machines. Click the link given below.

Source: https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm

3. Tasks

1. Create a new repository for this activity.
2. Install Docker and enable the docker socket.
3. Add to Docker group to your current user.
4. Create a Dockerfile to install web and DB server.
5. Install and build the Dockerfile using Ansible.
6. Add, commit and push it to your repository.

4. Output (screenshots and explanations)

Step 1:

Install the docker for the workstation and control node using "sudo apt install docker.io"

```
erwin@server2:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Step 2:

Check if the docker has been properly installed using docker –version

```
erwin@workstation:~$ docker --version
Docker version 27.3.1, build ce12230
```

Step 3

Create a group named docker and add the user to it

```
erwin@server1:~$ sudo usermod -aG docker $USER
[sudo] password for erwin:
erwin@server1:~$ newgrp docker
```

Step 6:

Create ansible playbook with the following code

```yaml
---
- name: Build Docker File
  hosts: localhost
  become: yes
  tasks:
    - name: build image from dockerfile
      community.docker.docker_image:
        source: build
        build:
          path: /home/erwin/Act11__Ballesteros
          dockerfile: dockerfile.web
        name: db_web_server
        tag: latest
        push: no
```

Step 7: Create a dockerfile with the following code

```
  GNU nano 7.2                                    dockerfile.web
#using UBUNTU
FROM ubuntu:latest

#install updates
RUN apt-get update && apt-get install -y \
        apache2 \
        mysql-server \
        && apt-get clean

#start services
CMD service mysql start && service apache2 start && tail -f /dev/null

#expose necessary poorts
EXPOSE 80 3306
```

Step 8: Run the ansible playbook

```
erwin@workstation:~/Act11__Ballesteros$ ansible-playbook --ask-become-pass run.yml
BECOME password:

PLAY [Build Docker File] ***********************************************************************

TASK [Gathering Facts] *************************************************************************
ok: [localhost]

TASK [build image from dockerfile] *************************************************************
changed: [localhost]

PLAY RECAP *************************************************************************************
localhost                  : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    i
gnored=0

erwin@workstation:~/Act11__Ballesteros$ docker images
REPOSITORY      TAG       IMAGE ID       CREATED         SIZE
db_web_server   latest    5cf14a9fb424   2 minutes ago   751MB
ubuntu          latest    59ab366372d5   4 weeks ago     78.1MB
hello-world     latest    d2c94e258dcb   18 months ago   13.3kB
```

Step 9: Verify if the docker image has been created properly by typing “docker images”

```
erwin@workstation:~/Act11__Ballesteros$ docker images
REPOSITORY      TAG       IMAGE ID       CREATED          SIZE
db_web_server   latest    5cf14a9fb424   19 minutes ago   751MB
ubuntu          latest    59ab366372d5   4 weeks ago      78.1MB
hello-world     latest    d2c94e258dcb   18 months ago    13.3kB
```

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?

One of the benefits of containerization is that it is consistent compared to virtualization because they only run within the created environment. They are also very lightweight and do not need excessive resources to run. Many complex application relies on containerization as it allows consistent runtime environments

Conclusions:

In this activity we have performed the basic installation, and setting the groups and roles of the user in docker. We have also implemented the basic creation of a dockerfile and its automation when developing it into a docker image. This basic implementation will allow us to create more complex automation when deploying docker images

Github Link : https://github.com/Moznaim/Act11__Ballesteros